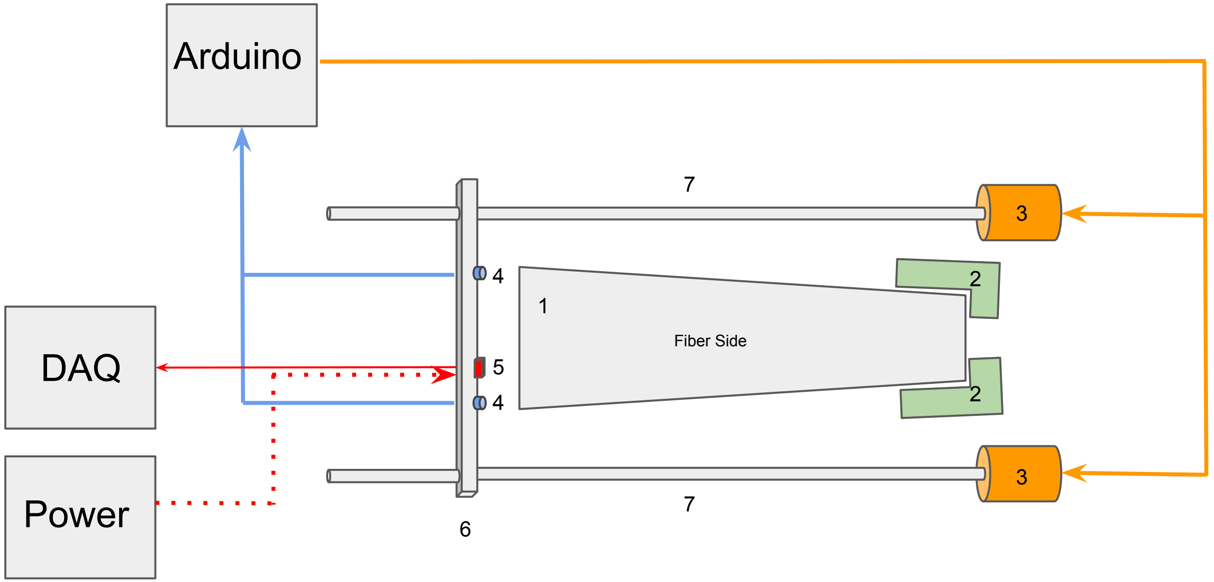
HCal QC Test Box

Sebastian Vazquez and Ron Belmont

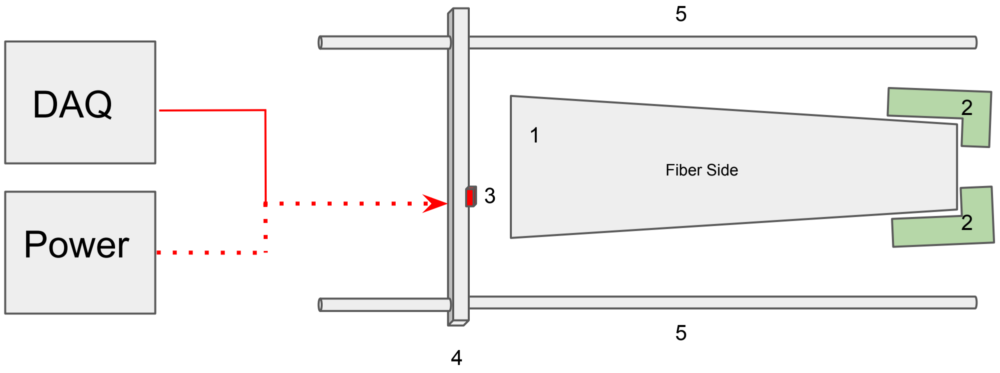
Inside the Box



1. HCal Tile
2. Pre-cut stencil for inner or outer tiles
3. Stepper Motor
4. Pressure Sensitive Button
5. SiPM
6. Extra large mounting bracket
7. Stepper Motor Arm

I assume that we will want to test both sides of the tile. Will we want to include two arrays of LEDs so the panel can be left in place or do we want to flip the tile to scan each side (requires two SiPMs)?

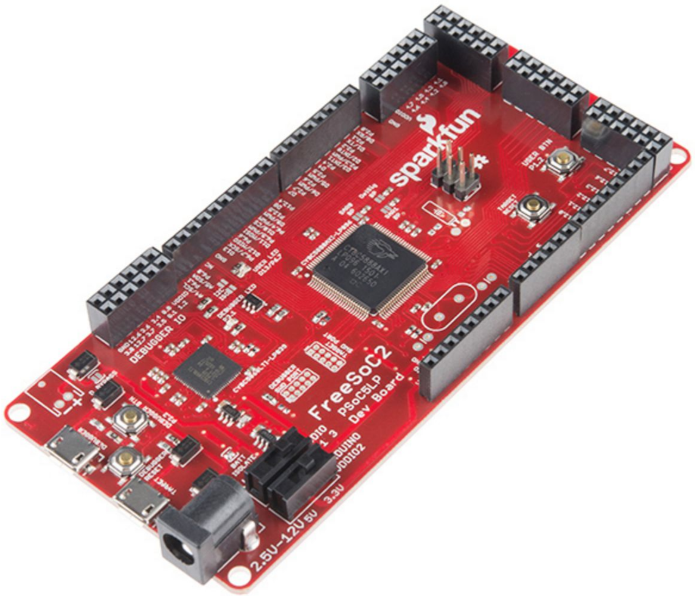
A Slightly More Simple Design



1. HCal Tile
2. Pre-cut stencil for inner or outer tiles
3. SiPM
4. Extra large mounting bracket
5. Mounting Bracket Guides

Here the mounting bracket is slid into place manually, we could have some sort of elastic or spring that applies continuous pressure to keep the SiPM firmly attached.

DAQ: FreeSoC2 ~ $49

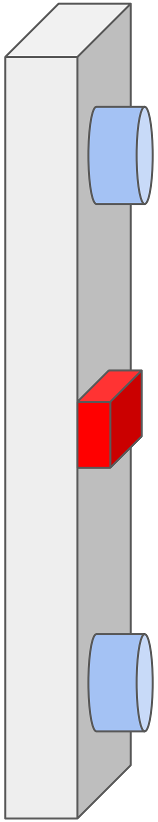


- CY8C5868LTI-LP039 & CY8C5888AXI-LP096 Cortex-M3 Processor Cores
- The target is in a TQFP-100 package which provides 72 IO pins versus the debugger’s QFN-68 package and 48 IO pins, and the target can operate at 80MHz versus the debugger’s 67MHz limit.
- 256kB of Flash Memory (each Core)
- 64kB of SRAM (each Core)
- 2kB of EEPROM (each Core)
- Arduino Uno R3-type Header
- Comes with its own IDE in which you can use C++ or a graphical interface to program in the chips functionality.

Another option might be a Xilinx FPGA (mentioned by J. Lajoie)

The FreeSoC2 could also be used to control the LEDs, any arduino model would work as well.

Smart Mounting



We envision this being essentially the current SiPM mounting bracket except longer with recessed possitions for housing two buttons.

The buttons act as an interrupt for the two stepper motors - stopping the beefy bracket when it is a certain distance from the tile edge.

Mini push button switches available through sparkfun for $0.35 each.

Coding the interrupt would be very easy with most microcontrollers.

This can be ignored if the manual mounting system is used

Less Smart Mounting

Depending on what stepper motors you use, you could have two settings (for the two tile shapes) and the two settings send the beefy bracket to a set position (ie the edge of the tile).

This design would eliminate the need to design a new mounting bracket (we could just make the bracket longer in Solid Works), but would not guarantee that the SiPM is always the same distance from the edge of the pannel.

This can be ignored if the mannual mounting system is implemented

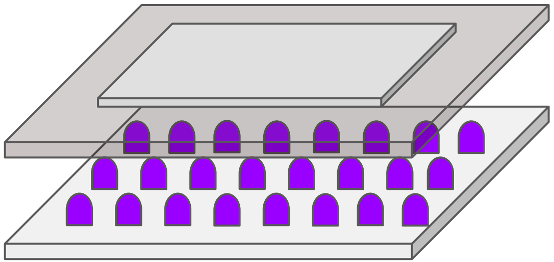
LEDs (405 nm) ~ 0.60_{ea}$

A FreeSoC2 microcontroller has 72 IO pins that can “operate” at 80MHz at 1V_{pp}, the LED pulse width could be tuned in 12.5 ns increments.

Programing the flash sequence if we can achieve the desired LED pulse width with the board would be very easy as we can write to the FreeSoC2 using C++.

LEDs would be mounted in a grid with collimator holes drilled through the support.

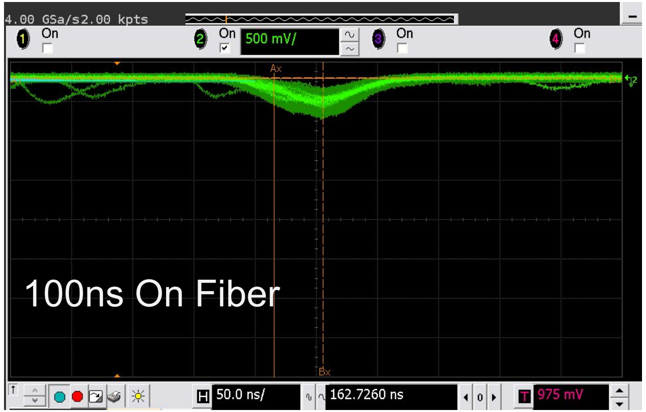
If we wanted a coarser resolution for more complete tile coverage, the support could be translucent without any form of collimation.

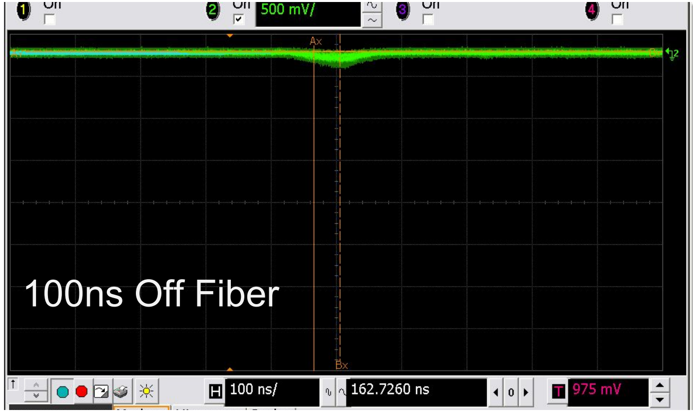


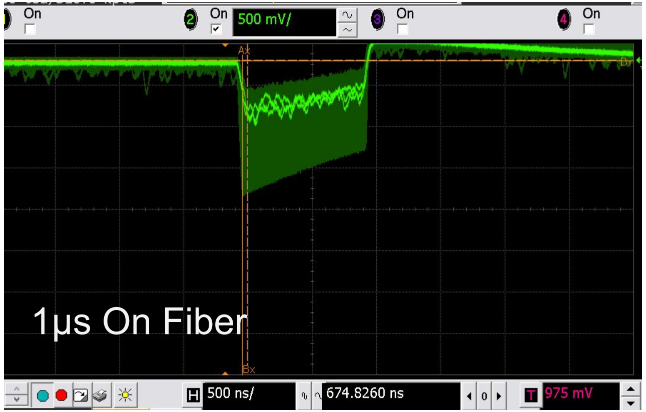
The greatest difficulty with operating the LEDs would be finding a pre-built control circuit that also has enough output pins for us to achieve the LED resolution we desire.

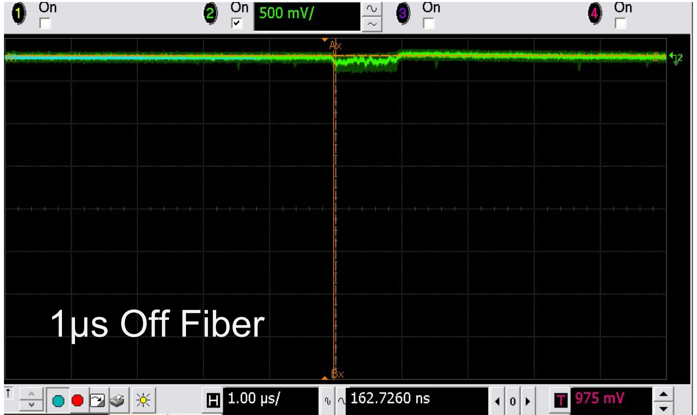
Exploring the Effect of a Wider LED Pulse (>50ns)

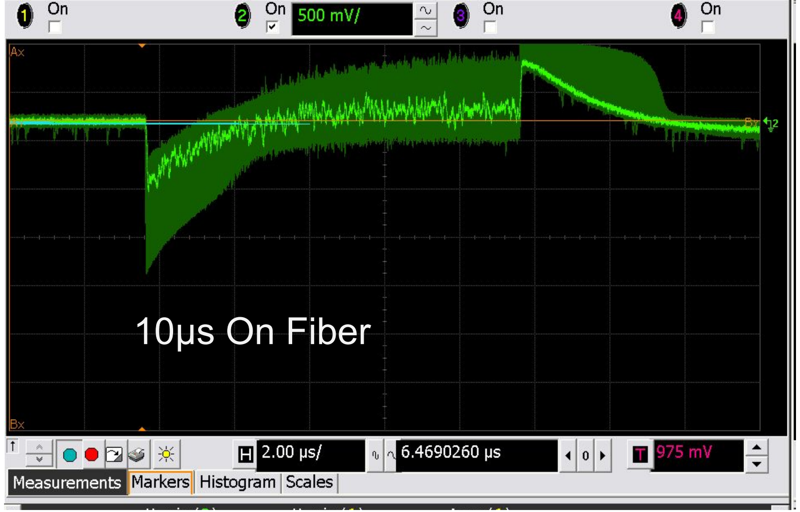
If we can still get meaningful LED scan data using a longer pulse width, we might be able to get by using a DAQ with lower sampling rate. In current scans, the LED pulse width is 50 ns.

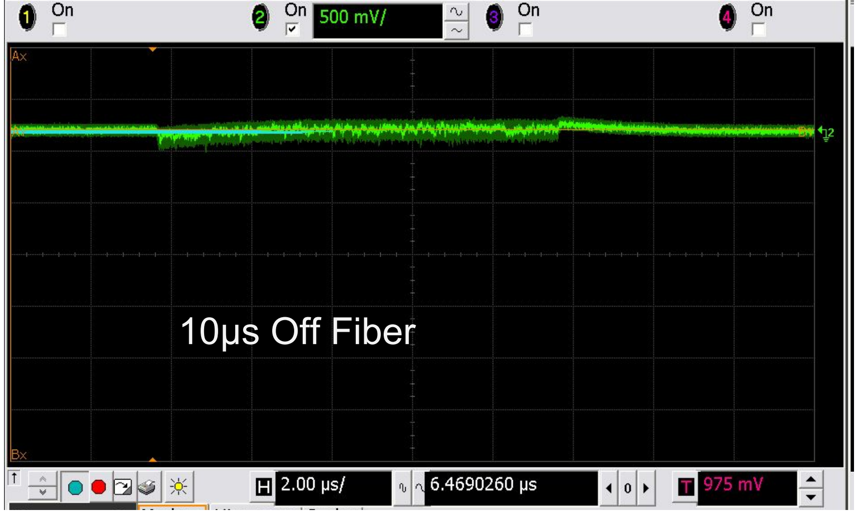


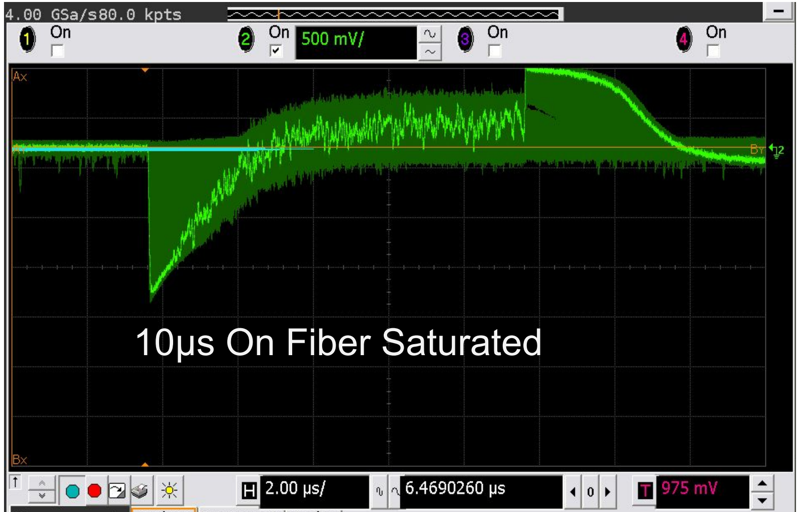












We will run a full panel LED scan to further examine the effect increased LED pulse width has on our current tile characterization.